Amendments to the Specification

Please insert the following paragraphs on page 1 of the specification beneath the title:

CROSS REFERENCE TO RELATED APPLICATIONS

This application is continuation of pending PCT Application No. PCT/GB99/03168, filed October 11, 1999, which is incorporated by reference in its entirety herein, and claims priority to U.S. Provisional Patent Application No. 60/115,952 filed on January 14, 1999, now abandoned, which is incorporated by reference in its entirety herein, and claims priority to GB Patent Application No. 9822075.9, filed October 10, 1998, now abandoned, which is incorporated by reference in its entirety herein.

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

Please insert the following paragraph on page 1 before the paragraph beginning on line 10 of the specification:

2. DESCRIPTION OF RELATED ART

Please insert the following paragraph on page 2 before the paragraph beginning on line 17 of the specification:

SUMMARY

The following is a summary of various aspects and advantages realizable according to various embodiments of the invention. It is provided as an introduction to assist those skilled in the art to more rapidly assimilate the detailed discussion of illustrative embodiments which ensues and does not and is not intended in any way to limit the scope of the claims which are appended hereto in order to particularly point out the invention.

Please amend the paragraph beginning on page 2, line 17, of the specification to read as follows:

~~It is an object of a~~ A first aspect of the present invention ~~to~~ provide<u>s</u> a method of generating an intermediate representation of program code, the method comprising the computer implemented steps of:

Please amend the paragraph beginning on page 2, line 30, of the specification to read as follows:

Also according to ~~the first~~ another aspect of the invention there is provided a method for generating an intermediate representation of computer program code written for running on a programmable machine, said method comprising:

Please amend the paragraph beginning on page 3, line 14, of the specification to read as follows:

According to another aspect of the present invention only a single register object need be generated to represent a given abstract register (which is preferable done for all abstract registers at initilisation), the state of each abstract register being defined by the expression objects referenced by the corresponding register object. Where more than one expression object is referenced by a given register object a "tree" of expression objects is generated having the register object as its 'root'. The expression trees referenced by each of the register objects will together form an "expression forest".

Please amend the paragraph beginning on page 3, line 22, of the specification to read as follows:

An advantage ~~of the invention~~ realizable according to the teachings herein is that any given expression object may be referenced to more than one register, and consequently an expression which is used by several different registers is not required to be created and assigned to each of those registers separately, but may be created once and referenced to each of the registers. In other words, expression trees may be linked together by expression objects which are referenced by more than one register object. Thus, a given expression object may be common to a number of expression trees within the expression forest.

Please amend the paragraph beginning on page 3, line 29, of the specification to read as follows:

By avoiding making multiple copies of the same expression, the invention reduces the time required to create the intermediate representation, and reduces the memory space occupied by the intermediate representation.

Please amend the paragraph beginning on page 4, line 1, of the specification to read as follows:

A further advantage ~~of the present invention~~ realizable according to the teachings herein is that expressions that become redundant can be very efficiently identified. When a new expression is assigned to a register object any expression previously referenced by that register object becomes redundant, except insofar as it is referenced by other register objects. These multiple references are detected using reference counting, described below.

Please amend the paragraph beginning on page 4, line 30, of the specification to read as follows:

According to ~~a second~~ another aspect of the present invention there is provided a method for generating an intermediate representation of computer code written for running on a programmable machine, said method comprising:

Please amend the paragraph beginning on page 5, line 9, of the specification to read as follows:

According to ~~the second~~ another aspect of the present invention there is provided a method of generating an intermediate representation of program code expressed in terms of the instruction set of a subject processor comprising at least one variable sized register, the method comprising the computer implemented steps of:

Please amend the paragraph beginning on page 7, line 5, of the specification to read as follows:

The ~~second aspect of the present invention overcomes~~ foregoing approach enables overcoming a problem which arises during emulation of a processor, and specifically when the emulated processor utilises variable sized registers. The nature of the problem addressed is best appreciated by example.

Please amend the paragraph beginning on page 9, line 18, of the specification to read as follows:

The use ~~in accordance with the invention~~ of separate abstract registers to represent each of the possible sizes of subject processor registers as described above, is advantageous because it allows data to be written to or moved from an abstract register in the intermediate representation without requiring extra processing during a region of a program which uses only one width of data. ~~The invention only requires~~ Thus, a calculation [[to]] only need be made (ie. the combination of data of different widths) on those infrequent occasions when the intermediate representation is required to represent data of different widths being written to and read from a subject processor register.

Please amend the paragraph beginning on page 9, line 26, of the specification to read as follows:

~~A third~~ Yet another aspect of the present invention reduces the amount of translated code. It is a property of subject code that:

Please amend the paragraph beginning on page 10, line 1, of the specification to read as follows:

According to ~~a third~~ another aspect of the present invention, there is provided a method of generating an intermediate representation of computer program code, the method comprising the computer implemented steps of:

Please amend the paragraph beginning on page 10, line 13, of the specification to read as follows:

~~The third aspect of the present invention~~ Such approaches reduce[[s]] the amount of translated code by permitting multiple, but simpler, blocks of intermediate representation code for single Basic Blocks of subject code. In most cases only one simpler translated block will be required.

Please amend the paragraph beginning on page 10, line 17, of the specification to read as follows:

According to another aspect of the present invention, there is provided a method for generating an intermediate representation of computer code written for running on a programmable machine, said method comprising:

Please amend the paragraph beginning on page 11, line 21, of the specification to read as follows:

This ~~aspect of the invention~~ approach may be applied to static translation, but is particularly applicable to emulation via dynamic binary translation. According to the invention, an emulation system may be configured to translate a subject processor program Basic Block by Basic Block. When this approach is used, the state of an emulated processor following execution of a Basic Block of program determines the form of the IR Block used to represent a succeeding Basic Block of the program.

Please amend the paragraph beginning on page 12, line 11, of the specification to read as follows:

A further advantage ~~of the third aspect of the invention~~ is that the resulting IR Blocks and IsoBlocks of intermediate representation are less complex than an intermediate representation which is capable of representing all entry conditions, and may therefore be optimised more quickly and will also be translated into target processor code which executes more quickly.

Please amend the paragraph beginning on page 12, line 16, of the specification to read as follows:

~~The third aspect of the present invention~~ This approach also exploits subject code instructions which may have a number of possible effects or functions, not all of which may be required when the instruction is first executed, and some of which may not in fact be required at all. This aspect of the invention may only be used when the intermediate representation is

generated dynamically. That is, [[the]] a preferred method according to the present invention preferably comprises, when the intermediate representation of the program is generated dynamically as the program is running, the computer implemented steps of:

Please amend the paragraph beginning on page 14, line 14, of the specification to read as follows:

Although the ~~third aspect of the invention as~~ approach just described above relates to the generation of intermediate representation, the steps described therein may be applied to the generation of target code directly from subject code, without the generation of intermediate representation.

Please amend the paragraph beginning on page 14, line 18, of the specification to read as follows:

Thus, the present invention may also provide[[s]] a method of generating target code representation of computer program code, the method comprising the computer implemented steps of:

Please amend the paragraph beginning on page 15, line 1 of the specification to read as follows:

According to ~~a fourth~~ another aspect of the present invention there is provided a method of dynamically translating first computer program code written for compilation and/or translation and running on a first programmable machine into second computer program code for running on a different second programmable machine. Said method comprising:

Please amend the paragraph beginning on page 15, line 15 of the specification to read as follows:

~~The present invention~~ This method realises the benefits of using intermediate representation in the real time translation of computer code.

Please amend the paragraph beginning on page 15, line 17 of the specification to read as follows:

Brief Description of the Drawings

An illustrative specific embodiment of the present invention applied to a dynamic emulation system will now be described, by way of example only, with reference to the accompanying drawings, in which:

Please amend the paragraph beginning on page 15, line 29 of the specification to read as follows:

Detailed Description of Illustrative Embodiments

The illustrative embodiments of the invention described below [[is]] provide, among other aspects, a system for emulating the instruction set of one processor on a processor of a different type. In the following description the term subject processor refers to a processor which is to be emulated by an emulation system and target processor refers to a processor upon which the emulation system is run. The system is a dynamic binary translation system which essentially operates by translating Basic Blocks of instructions in the subject processor code into target processor code as they are required for execution. The emulation system, as described below, comprises three major components, referred to respectively as a Front End, a Core, and a Back End. The subject processor instructions are decoded and converted into the intermediate representation

by the Front End of the emulation system. The Core of the emulation system analyses and optimises the intermediate representation of the subject processor instructions, and the Back End converts the intermediate representation into target processor code which will run on the target processor.

Please amend the paragraph beginning on page 18, line 21 of the specification to read as follows:

A series of illustrated examples will be used to convey how the emulation system uses expression objects (referred to as Expressions) and abstract registers to build up an intermediate representation of subject processor instructions. Figures 1 to 5 show step by step (with the progression of incremental steps indicated by reference numerals 1-12 in Figures 1 to 5), how the following pseudo-assembler code is represented in the Core using abstract registers:

Please amend the paragraph beginning on page 26, line 14 of the specification to read as follows:

If the initial state of a subject processor register on entry to a Basic Block were to be unknown at translate time, target-processor code to test the state of the register would have to be generated. For this reason, the emulation system according to the invention ensures that the state of each subject processor register is always known at translate time. In the system according to the present invention this is done by propagating the register state from one Intermediate Representation (IR) Block to the next. For example, IR Block 100 propagates the state of 'd0' to its successor IR Block 200, and IR Block 200 acts in a similar way propagating register state to IR Block 300. An example of this propagation of the subject processor register state is shown in Figure 6.

Please amend the paragraph beginning on page 26, line 24 of the specification to read as follows:

In Figure 6, IR Block 200 has two possible successors, either IR Block 300 or back at the beginning of IR Block 200. The route between IR Blocks 200 and 300 is shown with an arrow labeled as 'a'. The route from the end back to the beginning of IR Block 200 is shown as a dotted line labeled 'b' (a dotted line is used since, although this route exists it has not yet been traversed in the current execution of the translated program). If during the execution of the translated program, IR Block 200 were to branch back to itself along route 'b', the states it propagates would be incompatible with the abstract registers states which were originally passed to IR Block 200 by IR Block 100. Since the intermediate representation is specific to the state of the abstract registers IR Block 200 cannot be re-executed. For the correct operation of the invention across IR Block boundaries, each IR Block must have an *unambiguous* representation of the current state of the subject processor register (as represented by the abstract registers). The existence of route 'b' therefore is incompatible with the operation of the invention across the boundary between IR Block 100 and IR Block 200.

Please amend the paragraph beginning on page 27, line 6 of the specification to read as follows:

To overcome this problem the invention is able to represent a Basic Block of subject processor code using more than one IR Block with different entry conditions. The IR Blocks which are used to represent a single Basic Block with different entry conditions are referred to as IsoBlocks. Each IsoBlock is a representation of the same Basic Block of subject processor code, but under different entry conditions. Figure 7 shows two IsoBlocks which are used to overcome the problem illustrated in Figure 6. IsoBlock [[2a]] 200a is a correct representation of Basic Block 2, but only if the state of subject processor register 'd0' at the start of IR Block 200 is ✓XX (this corresponds to IR block 200 of Figure 6). When successor route 'b' in Figure 7 is

traversed for the first time, all the IsoBlocks in existence which represent Basic Block 2, (there is only one in this case, the IR Block), are tested for compatibility with the abstract register states that are to be propagated (i.e. ✓✓X). If a compatible IsoBlock is found (i.e. one that begins with the register state ✓✓X), the successor route 'b' will be permanently connected to that IsoBlock. In the illustrated example of Figure 7 there is no existing IsoBlock that route 'b' is compatible with, and so new IsoBlock [[2b]] 200b, must be created. IsoBlock [[2b]] 200b is created by decoding for a second time the subject processor instructions that make up Basic Block 2, using an initial assumption that the state of subject processor register 'd0' at the start of Basic Block 2 is ✓✓X.

Please amend the paragraph beginning on page 27, line 24 of the specification to read as follows:

When successor route 'c', originating from IsoBlock [[2b]] 200b, is traversed for the first time, a compatibility test is performed with IR Block 300. Since route 'c' is compatible with IR Block 300, a new IsoBlock does not need to be created, and both successor route 'a' and successor route 'c' are connected to IR Block 300.

Please delete page 29 entitled "Related Applications" in its entirety.

Please insert the following heading on page 30 before line 1 of the specification:

CLAIMS: